

PATENT ABSTRACTS OF JAPAN

(11) Publication number : 2002-312171
 (43) Date of publication of application : 25.10.2002

(51) Int.Cl.

G06F 9/445
 G06F 3/00
 G06F 11/26
 G06F 11/28

(21) Application number : 2001-110221

(22) Date of filing : 09.04.2001

(71) Applicant : MATSUSHITA ELECTRIC IND CO LTD

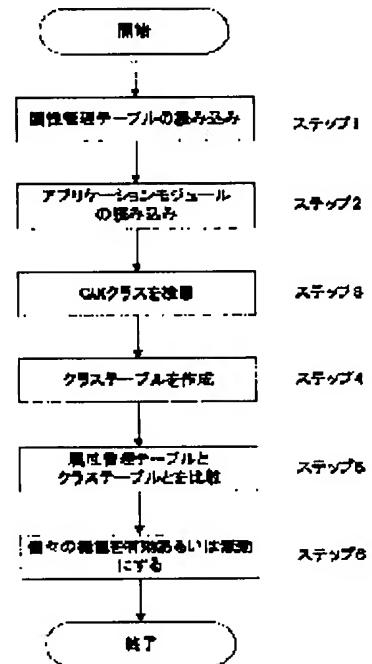
(72) Inventor : OKADA YUJI
 YAMAGUCHI TAKEHITO
 HISATOMI KENJI
 MATSUNAGA SHIGEKI

(54) HARDWARE-CONTROL SOFTWARE AND DEVELOPMENT SUPPORT SYSTEM THEREOF

(57) Abstract:

PROBLEM TO BE SOLVED: To efficiently develop software running on hardware having different attributes.

SOLUTION: A hardware-control software is made, with the attributes of a plurality of kinds of hardware taken into consideration. When information on the attributes of a piece of hardware on which the software runs is acquired from an attribute management table, the individual functions possessed by the software are each made valid or invalid, based on the information on the attributes of the hardware. It does not matter whether the individual functions are enabled/disabled by the software itself or by a control system on the hardware, on which the software runs.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-312171

(P2002-312171A)

(43)公開日 平成14年10月25日(2002.10.25)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
G 0 6 F	9/445	G 0 6 F	6 5 1 E 5 B 0 4 2
3/00	6 5 1	11/26	5 B 0 4 8
11/26		11/28	L 5 B 0 7 6
11/28		9/06	6 5 0 B 5 E 5 0 1

審査請求 未請求 請求項の数4 O.L (全12頁)

(21)出願番号	特願2001-110221(P2001-110221)	(71)出願人	000005821 松下電器産業株式会社 大阪府門真市大字門真1006番地
(22)出願日	平成13年4月9日(2001.4.9)	(72)発明者	岡田 雄治 大阪府門真市大字門真1006番地 松下電器 産業株式会社内
		(72)発明者	山口 岳人 大阪府門真市大字門真1006番地 松下電器 産業株式会社内
		(74)代理人	100083172 弁理士 福井 豊明

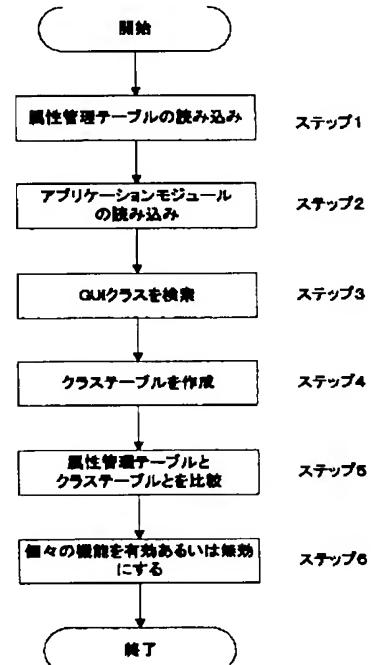
最終頁に続く

(54)【発明の名称】 ハードウェア制御ソフトウェア及びその開発支援システム

(57)【要約】

【課題】 異なる属性のハードウェア上で動作するソフトウェアを効率よく開発すること。

【解決手段】 複数種のハードウェア属性を考慮してハードウェア制御ソフトウェアを作成しておく。そして、当該ソフトウェアを動作させるハードウェアの属性情報を属性管理テーブルから取得したとき、このハードウェア属性情報に基づいて当該ソフトウェアが保持する個々の機能を有効あるいは無効にする。なお、上記個々の機能を有効あるいは無効にするのは、当該ソフトウェア自身、当該ソフトウェアを動作させるハードウェア上の制御システムのいずれであってもかまわない。



【特許請求の範囲】

【請求項1】 複数種のハードウェア属性を考慮して作成されたハードウェア制御ソフトウェアであって、当該ソフトウェアを動作させるハードウェアの属性情報を属性管理テーブルから取得したとき、このハードウェア属性情報に基づいて当該ソフトウェアが保持する個々の機能を有効あるいは無効にするスイッチ手段を備えたことを特徴とするハードウェア制御ソフトウェア。

【請求項2】 特定のハードウェアを制御する制御システムにおいて、当該システムのハードウェア属性情報を属性管理テーブルから取得したとき、このハードウェア属性情報に基づいて、複数種のハードウェア属性を考慮して作成されたハードウェア制御ソフトウェアが保持する個々の機能を有効あるいは無効にするスイッチ手段を備えたことを特徴とする制御システム。

【請求項3】 エミュレーターを使用することによって、特定のハードウェアを制御するソフトウェアの開発を支援する開発支援システムにおいて、

複数種のハードウェア属性を管理する属性管理テーブルと、
上記属性管理テーブルから取得した特定のハードウェア属性情報に基づいて、複数種のハードウェア属性を考慮して作成されたエミュレーターが保持する個々の機能を有効あるいは無効にするスイッチ手段と、
を備えたことを特徴とする開発支援システム。

【請求項4】 複数のソフトウェア部品を組み合わせることによって、特定のハードウェアを制御するソフトウェアを開発できるようにした開発支援システムにおいて、

複数種のハードウェア属性を管理する属性管理テーブルと、
上記属性管理テーブルから取得した特定のハードウェア属性情報に基づいてソフトウェア部品個々を有効あるいは無効にするスイッチ手段と、
を備えたことを特徴とする開発支援システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、特定のハードウェアを制御するソフトウェア、及び、その開発支援システムに関する。

【0002】

【従来の技術】 図1は、従来の典型的なデジタル複合機に組み込まれているソフトウェア2-1の構成図であり、以下その構成を動作とともに説明する。なお、デジタル複合機とは、スキャナ・プリンタ・ファックス等のデジタル画像機器をデジタル複写機に統合して省スペース化を実現しようとする装置をいう。

【0003】 まず、アプリケーションモジュール2-2がスキャン実行の指示をスキャナコントロールモジュー

ル2-3に出す(図14、ステップ①)。これによつて、スキャナコントロールモジュール2-3は、原稿をスキャンして画像をメモリに書き込み、この画像のメモリアドレスと共にスキャン終了の通知をアプリケーションモジュール2-2に返す(図14、ステップ②)。

【0004】 次いで、アプリケーションモジュール2-2は、受け取った画像のメモリアドレスと共にプリント実行の指示をプリンタコントロールモジュール2-4に出す(図14、ステップ③)。これによって、プリンタ

10 コントロールモジュール2-4は、受け取ったメモリアドレスから画像にアクセスしてプリントを実行した後、プリント終了の通知をアプリケーションモジュール2-2に返す(図14、ステップ④)。

【0005】 二枚目以降の原稿についても同様である(図14、ステップ⑤～⑧)。

【0006】

【発明が解決しようとする課題】 ここで、上記従来のソフトウェア2-1は、デジタル複合機本体に固定的に組み込まれる(すなわち、個々のハードウェア用に作成される)のが通常であったため、変更したり差し替えたりすることを前提に開発されていなかった。

【0007】 したがって、上記従来のソフトウェア2-1を作成(開発)する際には、スキャナやプリンタ等の制御について詳細に検討する必要があった。また、開発したソフトウェア2-1の動作確認を行うためには当該ソフトウェア2-1をハードウェア上で動作させることになるが、この動作確認には多大な工数が必要となるという問題があった。更に、ハードウェアが故障したり手元になかたりすると、ソフトウェア2-1の動作確認ができずデバッグが進まないという問題もあった。

【0008】 このようなことから、個々のハードウェア用にソフトウェア2-1を作成するのではなく、異なるハードウェア上でも動作するソフトウェア2-1を共通に作成することが提案されている。しかしながら、これを実現するためには、ハードウェアによって属性(仕様)が異なるという問題、更には、ハードウェアの属性によっては操作パネルから指定可能な項目が使用できない(すなわち、操作パネルでは指定可能な項目として表示されているものの、この項目に対応する機能がハードウェアに備わっていない)という問題を解決する必要がある。

【0009】 本発明は、上記従来の事情に基づいて提案されたものであって、異なる属性のハードウェア上で動作するソフトウェアを効率よく開発することを目的とする。

【0010】

【課題を解決するための手段】 本発明は、特定のハードウェアを制御するソフトウェアを前提としており、上記目的を達成するために以下の手段を採用している。

【0011】 すなわち、複数種のハードウェア属性を考

3
慮して上記ソフトウェアを作成しておく。そして、当該ソフトウェアを動作させるハードウェアの属性情報を属性管理テーブルから取得したとき、このハードウェア属性情報に基づいて当該ソフトウェアが保持する個々の機能を有効あるいは無効にする。このようにすれば、異なる属性のハードウェア上で動作するソフトウェアを効率よく開発することが可能である。

【0012】なお、上記個々の機能を有効あるいは無効にすることは、当該ソフトウェア自身、当該ソフトウェアを動作させるハードウェア上の制御システムのいずれであってもかまわない。

【0013】また、本発明は、エミュレーターを使用することによって、特定のハードウェアを制御するソフトウェアの開発を支援する開発支援システムに適用することも可能である。

【0014】すなわち、複数種のハードウェア属性を属性管理テーブルで管理しておく。これによって、スイッチ手段は、上記属性管理テーブルから取得した特定のハードウェア属性情報に基づいて、複数種のハードウェア属性を考慮して作成されたエミュレーターが保持する個々の機能を有効あるいは無効にする。このようにすれば、異なる属性のハードウェア上で動作するエミュレーターを効率よく開発することが可能である。

【0015】更に、本発明は、複数のソフトウェア部品を組み合わせることによって、特定のハードウェアを制御するソフトウェアを開発できるようにした開発支援システムに適用することも可能である。

【0016】すなわち、複数種のハードウェア属性を属性管理テーブルで管理しておく。これによって、スイッチ手段は、上記属性管理テーブルから取得した特定のハードウェア属性情報に基づいてソフトウェア部品個々を有効あるいは無効にする。このようにすれば、異なる属性のハードウェア上で動作するソフトウェアを効率よく開発することが可能である。

【0017】

【発明の実施の形態】以下に本発明の実施の形態を図面にしたがって詳細に説明する。

(第1の実施の形態) まず、デジタル複合機に組み込まれるソフトウェアの構成は上記従来の構成(図1参照)と同じであるが、本発明と従来技術とではアプリケーションモジュール2-2の機能が異なる。

【0018】すなわち、従来のアプリケーションモジュール2-2は、ユーザーからのジョブの依頼をコマンドに変換する機能と、スキャナコントロールモジュール2-3及びプリンタコントロールモジュール2-4に対してページ単位でコマンドを送ってその動作を制御するシーケンス制御機能とを備えていた。それに対し、本実施の形態では、このシーケンス制御機能をアプリケーションモジュール2-2から分離してスキャナコントロールモジュール2-3とプリンタコントロールモジュール2-

-4とに吸収させることで、アプリケーションモジュール2-2の役割を軽減するようしている。

【0019】すなわち、アプリケーションモジュール2-2は、スキャン実行の指示をスキャナコントロールモジュール2-3に出した後、プリント実行の指示をプリンタコントロールモジュール2-4に出す(図2、ステップ①→②)。これらスキャン指示とプリント指示とは順序が逆でも差し支えない。

【0020】これによって、スキャナコントロールモジュール2-3は、原稿をスキャンして画像をメモリに書き込み、スキャン終了の通知をアプリケーションモジュール2-2に返す(図2、ステップ③)。

【0021】一方、プリンタコントロールモジュール2-4は、上記のようにプリント指示を受けると、メモリに画像が存在するか否かを確認し、画像が存在すればプリントを実行する。それに対し、画像が存在しなければ未だスキャンが終わっていないことになるので、一定時間後再び画像が存在するか否かを確認する。そして、画像が存在することを確認できたらプリントを実行し、このプリントが終了するとプリント終了の通知をアプリケーションモジュール2-2に返す(図2、ステップ④)。

【0022】このようにアプリケーションモジュール2-2の機能の一部をスキャナコントロールモジュール2-3とプリンタコントロールモジュール2-4とに吸収させる代わりに、アプリケーションモジュール2-2を機能別に分割するようにしてもよい。すなわち、図3に示すように、上記アプリケーションモジュール2-2をスキャナシーケンス制御モジュール3-3とプリンタシーケンス制御モジュール3-4とその残余であるアプリケーションモジュール3-2とに分割しても同様の効果が得られる。

【0023】以上のように、本実施の形態では、スキャナの制御とプリンタの制御とを独立して行うようしているため、スキャン完了通知を受け取ってからプリント指示を出すといった制御の順序を管理する必要がなくなる。これによって、アプリケーションモジュール2-2(3-2)の機能が単純になることはいうまでもない。

【0024】また、従来の構成では、スキャン時に動作中断原因(例えば紙ジャム)が発生すると、スキャナコントロールモジュール2-3が紙ジャム発生通知をアプリケーションモジュール2-2に出すようになっていた。そして、ユーザーが紙をセットし直して再スタートを指示すると、再開可能な状態になったことを検知したアプリケーションモジュール2-2が再びスキャン実行通知をスキャナコントロールモジュール2-3に出す。すなわち、従来の構成によると、動作中断時にアプリケーションモジュール2-2の処理が必要であった。

【0025】それに対し、本実施の形態によると、アプリケーションモジュール2-2はスキャンの指示やプリ

ントの指示を一旦出すと、その後は完了通知を受け取るまで一切スキャナコントロールモジュール2-3とプリンタコントロールモジュール2-4とに関与しない。すなわち、スキャナコントロールモジュール2-3・プリンタコントロールモジュール2-4それぞれが動作中断を検知するとともに、再開可能な状態になったときに当該動作を再開するようになっている。このようにすれば、アプリケーションモジュール2-2の開発工数が少なくなるのはいうまでもない。

【0026】ところで、アプリケーションモジュール2-2とスキャナコントロールモジュール2-3あるいはプリンタコントロールモジュール2-4とは、APIと呼ばれる広く知られた方法で結合して動作するようになっている。このAPI（アプリケーション・インターフェイス）とは、複数のソフトウェアモジュールが連携して動作する際に用いられるものであって、一方のモジュールが処理を実行するコマンド名を公開し、他方のモジュールがこのように公開されたコマンド名を使用してソフトウェアをプログラムする方法をいう。

【0027】上記APIを用いれば、一つのサーバーに複数のデジタル複合機が接続されているような場合に利便性が高まる。すなわち、上記スキャナコントロールモジュール2-3と上記プリンタコントロールモジュール2-4とはデジタル複合機に予め保持しておく。一方、上記アプリケーションモジュール2-2はサーバーに保持するようにし、ネットワーク経由でデジタル複合機から取得要求されたとき、当該アプリケーションモジュール2-2をサーバーからデジタル複合機に移動して実行する。

【0028】このような構成によると、アプリケーションモジュール2-2の管理（バージョンアップ等）は、サーバーに保持しているアプリケーションモジュール2-2に対してのみ行えばよい。すなわち、従来のように、すべてのデジタル複合機が保持しているアプリケーションモジュール2-2を管理する必要がなくなるので利便性が高まる。

（第2の実施の形態）上記したように、本発明によると、スキャナコントロールモジュール2-3がページ単位で画像をメモリに書き込む動作と、プリンタコントロールモジュール2-4がページ単位で画像をメモリから読み出す動作とは独立することになる。このような手法によると、1部コピーを行った場合などジョブが一つしか存在しない場合に問題が生じることはない。しかしながら、複数部コピーを行った場合などジョブが二つ以上存在する可能性のある場合は、以下に説明するようにジョブの混乱が生じる。

【0029】すなわち、スキャン動作が完了した後にプリント動作だけが継続中であるときは、スキャナが空いていることになるため引き続き次の原稿をコピーすべくスキャン動作が開始される場合がある。これによってジ

ョブが二つ以上存在することになるが、この場合、1ページ目のスキャン画像と2ページ目のスキャン画像のいずれをプリントすべきなのかプリンタコントロールモジュール2-4は区別が付かない。

【0030】このような問題を解決するには、図4に示すように、ジョブを識別するための「ジョブID」フィールド、メモリ上の読み出し開始アドレスを示す「開始アドレス」フィールド、スキャン中であるかスキャン完了であるかを示す「状態」フィールド、「スキャン完了ページ数」フィールド、「プリント完了ページ数」フィールド等からなるジョブ管理テーブルT1を当該装置のメモリ内に作成すればよい。

【0031】すなわち、アプリケーションモジュール2-2は、最初のコピー時にジョブ1のスキャンとしてスキャン実行指示をスキャナコントロールモジュール2-3に出すとともに、このジョブ1のプリント実行指示をプリンタコントロールモジュール2-4に出す。これによって、スキャナコントロールモジュール2-3は、スキャンした画像をメモリに書き込むと同時に、そのジョブ情報（ジョブIDは1）をジョブ管理テーブルT1に書き込む。

【0032】このようにすれば、プリンタコントロールモジュール2-4は上記ジョブ管理テーブルT1を参照することによって正しいメモリアドレスにアクセスできることになるため、ジョブの混乱は生じない。

（第3の実施の形態）従来、デジタル複合機の動作状態（終了ページ数やトナー残量など）は、図5に示すように、アプリケーションモジュール2-2がスキャナコントロールモジュール2-3あるいはプリンタコントロールモジュール2-4から受け取って操作パネルPに表示していた。

【0033】しかしながら、上記第1および第2の実施の形態では、スキャナコントロールモジュール2-3とプリンタコントロールモジュール2-4とがそれぞれ独立に動作し、その間、アプリケーションモジュール2-2は関与しない構成を採用している。したがって、上記従来のように、アプリケーションモジュール2-2がスキャナコントロールモジュール2-3あるいはプリンタコントロールモジュール2-4から上記動作状態情報を受け取ることができない。

【0034】このような問題を解決するには、図6に示すように、スキャナコントロールモジュール2-3とプリンタコントロールモジュール2-4とがアプリケーションモジュール2-2を介すことなく操作パネルPに上記動作状態情報を直接表示すればよい。このようにすれば、アプリケーションモジュール2-2は、アプリケーションの状態だけを操作パネルPに表示すればよく、スキャナコントロールモジュール2-3の状態やプリンタコントロールモジュール2-4の状態が変化しても何ら関与する必要がない。

【0035】このように各モジュールが操作パネルPに情報を直接表示する場合は、図7に示すように、スキャナコントロールモジュール2-3用・プリンタコントロールモジュール2-4用・アプリケーションモジュール2-2用として操作パネルPの画面領域を3つの領域E1・E2・E3に分割し、それぞれの領域E1・E2・E3に独立して情報を表示する構成としておく。

【0036】もちろん、この表示態様は単なる一例であり、別の表示態様を採用することも可能である。すなわち、操作パネルPがアプリケーションモジュール2-2・スキャナコントロールモジュール2-3・プリンタコントロールモジュール2-4の3モジュールすべてから表示情報を受け取ってメモリに保持し、そのうち1モジュールの情報のみを全領域に表示するようにしてもよい。

【0037】この場合、他の2モジュールの情報はメモリに保持されているだけで表示されることになるが、当該操作パネルPに切り替えボタンを設けておけば各モジュールの情報を順次切り替えて表示できる。なお、このような表示切り替え技術は公知であるため、ここでは詳しい説明を省略する。

(第4の実施の形態) ところで、アプリケーションモジュール2-2の機能には、スキャン画像をプリンタでプリントするコピー機能だけでなく、プリンタと同様の構成で接続されているモデムからスキャン画像を外部に送信するFAX機能、更には、ネットワーク通信手段からスキャン画像を外部に配信する配信機能など様々な機能がある。したがって、特定の装置(以下「装置」を「ハードウェア」という場合がある)だけでしか使用しないアプリケーションモジュール2-2を作成するのは効率が悪い。すなわち、異なる属性のハードウェアでも共有できるようにアプリケーションモジュール2-2を作成すると開発効率が高まる。

【0038】例えば、白黒デジタル複合機用にコピー・アプリケーションモジュールを作成する必要があるとともに、カラーデジタル複合機用にもコピー・アプリケーションモジュールを作成する必要がある場合を考える。

【0039】この場合、白黒あるいはカラーという細部の機能は異なるが、コピーという動作の中で共通の機能が多い。したがって、白黒デジタル複合機とカラーデジタル複合機とでコピー・アプリケーションモジュール2-2を共有し、重複した機能を別個に作成することを避け開発工数を減らすというのは自然な発想である。

【0040】しかしながら、属性の異なるいずれのハードウェアでも正しく動作するようにアプリケーションモジュール2-2を作成するのは容易でない。例えば、両面プリント機能を備えたアプリケーションモジュール2-2を作成する場合は、両面プリントが可能なハードウェアか否かによって操作パネルPの表示項目を変える必要がある。すなわち、両面プリントが可能なハードウェ

アであれば、その操作パネルPに両面プリントの項目を表示する必要があるのに対し、両面プリントが不可能なハードウェアであれば、その操作パネルPに両面プリントの項目を表示すべきでない。

【0041】このような問題を解決するには、図8に示すように、ハードウェアの属性を登録した属性管理テーブルT2をハードウェア毎に設ければよい。すなわち、アプリケーションモジュール2-2は、あるハードウェアで動作する際、上記属性管理テーブルT2を参照することによって当該ハードウェアの属性に対応する機能を特定し、この機能に対応する項目のみを操作パネルPに表示するようになっている(詳細は後述する)。

【0042】ここで、ハードウェアの属性には、両面プリントのように機能的な属性もあれば解像度のように量的な属性もあるため、上記属性管理テーブルT2にハードウェアの属性を登録する場合はその表現方法が問題となる。すなわち、本実施の形態では、機能的な属性については「有るか無いか」で表現することとし、また、量的な属性については「数値」で表現することとしている(図8参照)。

【0043】なお、このようにハードウェアによって異なる属性の項目は両面プリントだけでなく、両面スキャン・各種画像処理・拡大縮小・サムネールプリント・ページ連写・色変換など数多い。

【0044】以下、あるハードウェア(I)はAとBとCとを備え、別のハードウェア(II)はAとDとを備えている場合を想定し、本実施の形態を更に詳しく説明する。なお、Aは両面プリント機能、またBは濃度調節機能というように、ここでいうA・B・C・Dはそれぞれハードウェアの属性を示す。

【0045】まず、A・B・Cの3機能を備えたアプリケーションモジュール2-2を作成した場合、このアプリケーションモジュール2-2は、ハードウェア(I)で動作する場合にはすべての機能を実現することができ、一方、ハードウェア(II)で動作する場合にはB・Cの機能が無駄になるとともにDの機能を実現することができない。したがって、ハードウェア(I)(II)両方で動作することを目的としたアプリケーションモジュール2-2は、A・B・C・Dすべての機能を備えるよう作成しておくとともに、最終的には、当該アプリケーションモジュール2-2の実装先となるハードウェアの属性と対応する機能のみを選択的に実行できるよう作成しておくのが好ましい。

【0046】ただし、この場合は、A・B・C・Dすべての機能を備えるように作成しておいたアプリケーションモジュール2-2を目的のハードウェアに実装する際(あるいは実装に先立って)、A・B・C・D4機能の内いずれの機能が目的のハードウェアの属性と対応する機能であるかチェックする必要がある。しかしながら、このようなチェックをするためにはアプリケーションモ

ジユール2-2のソースコードと属性管理テーブルT2の内容とを比較する必要があり、この比較作業を開発者が行うのは妥当でない。そこで、本実施の形態では、以下に説明するスイッチ手段をアプリケーションモジュール2-2に含めておくようにしている。

【0047】すなわち、あるアプリケーションモジュール2-2が目的のハードウエアに実装されると、このモジュール2-2に含まれるスイッチ手段が起動して属性管理テーブルT2を読み込む(図9、ステップ1)。この属性管理テーブルT2がハードディスクに保存されたファイルである場合は、当該ファイルをオープンしてその内容をメモリに読み出すことになる。

【0048】次に、スイッチ手段は、当該アプリケーションモジュール2-2を読み込む(図9、ステップ2)。なお、JAVA(R)言語(Sun Microsystems社が開発したオブジェクト指向のインターフリタ言語)でアプリケーションモジュール2-2が記述されている場合、当該アプリケーションモジュール2-2のコード形態にはソースコードとビルドされたバイトコードの2種類がある。したがって、この2種類のコードのいずれかを解読できるようにスイッチ手段を作成しておく必要がある。

【0049】更に、スイッチ手段は、上記のように読み込んだアプリケーションモジュール2-2がJAVA(R)言語で記述されている場合は当該アプリケーションモジュール2-2からGUIクラスを検索する(図9、ステップ3)。

【0050】GUI(Graphical User Interface)クラスとは、操作パネルPにアイコンや文字を表示する機能を備えたプログラムモジュールをいう。このGUIクラスによれば、例えば、両面コピーをするか否かを選択するために両面コピーという文字またはそれを表現した図柄の描かれたボタンを操作パネルPに表示することが可能となる。

【0051】なお、ここで使用する操作パネルPとしては、例えばタッチパネル機能を備えた液晶ディスプレイを採用するのが好ましい。また、ユーザーから両面コピーの指示があった場合はその旨をアプリケーションモジュール2-2が記憶し、このアプリケーションモジュール2-2がハードウエアを制御することによって当該コピーが両面コピーとなるようにしておく。

【0052】このようにタッチパネル機能を備えたディスプレイを使用してユーザーの意志を装置に反映する方法は広く知られている。例えば、JAVA(R)言語を使用して上述のGUIを実現する方法は株式会社祥泳社発刊の「独りJAVA(R)」(1999年5月25日発行)等に詳しい。

【0053】ところで、操作パネルP上に表示する各種機能をオン/オフするボタン形状のグラフィック・オブジェクトは、このグラフィック・オブジェクトを操作パ

ネルP上に表示するプログラム中、当該機能を司るクラスの名称と共に記述することができる。なお、クラスとは、オブジェクト指向プログラミングで機能や性質に共通点のあるオブジェクトをいう。

【0054】したがって、クラスの名称は、属性管理テーブルT2とクラステーブルとを比較(後述する)する必要上、属性管理テーブルT2で使用する属性の名称と一致させておく。例えば、属性管理テーブルT2で「両面プリント」という属性名称を使用する場合は、そのクラス(すなわち、操作パネルPにボタンを表示するプログラムにおける両面プリントボタンの表示機能を司るクラス)の名称も「両面プリント」としておけばよい。

【0055】さて、上記のようにGUIクラスを検索したスイッチ手段は、次いで、機能選択のオン/オフをボタンで表示する機能を司るクラスの名称を対象プログラムから検索し、図10に示すように、その名称と当該機能の有無などを対応付けたクラステーブルT3を作成する(図9、ステップ4)。そして、このクラステーブルT3と属性管理テーブルT2とを参照し、それぞれの名称とその機能有無とをパターンマッチングなどの方法で順次比較した後、当該アプリケーションモジュール2-2が保持する個々の機能のうち属性管理テーブルT2に有る機能を有効とする一方、属性管理テーブルT2に無い機能を無効とする(図9、ステップ5→ステップ6)。なお、このように機能を有効あるいは無効にする手法は後述する。

【0056】以上のように、本実施の形態によれば、異なる属性のハードウエアでも共有できるように作成されたアプリケーションモジュール2-2を目的のハードウエアに実装すると、このアプリケーションモジュール2-2が、当該ハードウエアの属性に対応する機能のみを備えるようにすることができる。

【0057】なお、ここではアプリケーションモジュール2-2にスイッチ手段を含めることとしているが、本発明はこれに限定されるものではない。すなわち、このようなスイッチ手段は、アプリケーションモジュール2-2の実装先となるハードウエア上の制御システム(例えば基本ソフトウェア)が備えてもよい。

【0058】また、アプリケーションモジュール2-2の機能を有効あるいは無効にするタイミングも上記したタイミングに限定されるものではない。すなわち、実装先となるハードウエアの属性が予めわかっている状況下では、この実装に先立って、例えばパソコン・コンピュータ上などでアプリケーションモジュール2-2の機能を有効あるいは無効にしておくことも可能である。

【0059】更に、「アプリケーションモジュール2-2に不要なクラスはどれだけあるか」あるいは「どのハードウエアでも共有できるようにアプリケーションモジュール2-2が作成されているか」といったことを容易に分析できるようにするために上記スイッチ手段が以下

の処理を行うようにしてもよい。

【0060】すなわち、上記のようにクラステーブルT3と属性管理テーブルT2とを比較したスイッチ手段は、属性管理テーブルT2とクラステーブルT3の両方に有る機能は「共通」、属性管理テーブルT2のみに有る機能は「属性管理テーブルのみ」、クラステーブルT3のみに有る機能は「クラステーブルのみ」と分類し、この分類結果を例えれば図11に示すようなフォーマットでプリンタ等から出力する。このようにすれば、アプリケーションモジュール2-2に不要なクラスはどれだけあるか、といったことが容易に分析できて便利である。

(第5の実施の形態) 上記第1～第4の実施の形態で説明した制御方法を用いるとアプリケーションモジュール2-2の開発工数が減るので、種々のサービスアプリケーションを開発して実行するというケースが出てくる。そこで、本実施の形態では、種々のサービスアプリケーションを開発した場合、その実行を簡便に行うための形態について説明します。デジタル複合機の一般的な機能はコピー機能とファックス機能であり、これらの機能は、図12(a)に示すように、操作パネルPで簡単に切り替え可能となっている。すなわち、コピー機能からファックス機能へ切り替えたい場合は、タッチパネルに表示されたファックスタブを選択すればよく、逆にファックス機能からコピー機能へ切り替えたい場合は、タッチパネルに表示されたコピータブを選択すればよい。

【0061】しかしながら、このような切り替え手法によると、更にスキャナやプリンタなどを使用するアプリケーションを追加した場合、切り替えたタブの数が多くなりすぎるという問題があった。

【0062】すなわち、デジタル複合機は複数のユーザーが共有して使用するのが通常であり、また、ユーザーによって使用するアプリケーションは異なる場合が多い。したがって、多数のアプリケーションがデジタル複合機に実装されている場合、これらアプリケーションに対応するだけのタブを単に並列に並べても、個々のユーザーにとっては選択肢が増えて煩雑になるばかりでメリットがない。

【0063】例えば、スキャン画像をOCR処理してその結果を電子メールで配信する画像配信アプリケーションと、スキャン画像をサーバーに登録する画像登録アプリケーションと、ネットワークを通じてインターネット情報にアクセスしてその情報をプリントするインターネット情報プリントアプリケーションとがデジタル複合機に実装されている場合、上記画像配信アプリケーションのみ使用するユーザーにとっては、他のアプリケーションに対応するタブ、すなわち画像登録タブ及びインターネット情報プリントタブは不要である。

【0064】そこで、本実施の形態では、各アプリケーションが共通メニューと個人メニューのいずれに属するか図示しない分類管理テーブルに登録しておくことと

し、デジタル複合機の制御システムがメニューを表示する際に当該分類管理テーブルを参照する構成を採用している。

【0065】これによって、上記制御システムは、メニューが共通メニューであることが判明した場合は当該メニューをタブに表示し、一方、メニューが個人メニューであることが判明した場合は当該メニューを図12(b)に示すような選択窓12-1に表示する。このようすれば、多数のアプリケーションを追加した場合であっても、切り替えたタブの数が多くなりすぎるという問題は生じない。

【0066】更に、デジタル複合機を使用する際に個人認証を行い、この認証結果に応じて個人メニューを表示する構成を採用すればより使いやすいものとなる。例えば、分類管理テーブルにユーザーIDの項目を追加しておくとともに、アプリケーションの利用頻度が高いユーザーIDを当該アプリケーションと関連づけて分類管理テーブルに登録しておく。そして、個人認証を行った後、当該認証がされたユーザーIDと関連づけられているアプリケーションについてのみ個人メニューを表示する。このようすれば、個人毎に異なる使用頻度に応じてアプリケーションをメニュー表示できることになるため、選択時の煩雑さがなくなり使いやすい。

【0067】なお、アプリケーションはデジタル複合機本体が必ずしも保持する必要はない。すなわち、サーバーとデジタル複合機とがネットワークで接続されている場合は、サーバーがアプリケーションモジュール2-2を保持する構成としてもよい。

【0068】このような構成によると、当該ネットワークに複数のデジタル複合機が接続されている場合でもサーバーにのみアプリケーションをインストールすればよいので利便性が高まる。すなわち、各デジタル複合機の制御システムは例えば電源投入後にアプリケーションモジュール2-2をサーバーからダウンロードして実行すればよい。このような仕組みは特開平7-93230号公報にも開示されており、また、JAVA(R)言語を使用することによって広く一般的に行われている。

【0069】ここで、例えばカラースキャナ及びカラープリンタを備えたデジタル複合機と、白黒スキャナ及び白黒プリンタを備えたデジタル複合機とをネットワークを介してサーバーに接続した場合は、それぞれに対応したアプリケーションモジュール2-2を用意しなければならない。これは、コピーという同種のアプリケーションモジュール2-2でも、カラーコピーと白黒コピーとでは操作パネルPにおける表示や機器制御時のパラメータの設定などが異なるためである。例えば、白黒コピーでは、「濃い」あるいは「薄い」という濃度の設定のみ存在するが、カラーコピーでは、色毎の濃度の設定だけでなく「青っぽく」あるいは「赤っぽく」といった設定が存在する。

【0070】しかしながら、このようにハードウェアの属性の数だけアプリケーションモジュール2-2を用意するのは、メモリ容量の面からみても作業工数の面からみても効率が悪い。

【0071】この問題を解決するには、カラーと白黒の両方の機能をアプリケーションモジュール2-2に含めるようにし、当該アプリケーションが、その実行時にハードウェアの属性に応じて処理を選択すればよい。ハードウェア属性の確認方法としては、ハードウェアとハードウェアコントロールモジュールとの通信で機種属性を取得するのが一般的である。

(第6の実施の形態) ところで、アプリケーションモジュール2-2を開発する際には、当該アプリケーションモジュール2-2を動作させるハードウェア上でのテストが必要になる。したがって、ハードウェアが故障したり手元になかったりすると、アプリケーションモジュール2-2の動作確認ができずデバッグが進まないという問題があった。そこで、このような場合は、ハードウェアの動作を仮想的に再現したハードウェアエミュレーションソフトウェア(以下「エミュレーター」という)を使用することが広く知られている。

【0072】ここで、少しずつ属性の異なる複数のハードウェア上で動作させる複数のアプリケーションモジュール2-2を開発する必要がある場合、これらハードウェア毎にエミュレーターを作成するのは効率が悪い。したがって、複数のハードウェアの動作をすべて網羅したエミュレーターを1つだけ作成することも考えられるが、このようなエミュレーターでは正しく動作テストをすることができない。テスト対象であるアプリケーションモジュール2-2を動作させるハードウェアが備えていない機能まで当該エミュレーターに含まれているためである。

【0073】そこで、本実施の形態では、複数のハードウェアの動作をすべて網羅したエミュレーターを作成した場合、このエミュレーターを特定のハードウェアに対応したエミュレーターに自動的に書き換えるプログラム(以下「スイッチ手段」という)を採用しており、以下の書き換え手順について説明する。なお、ここで採用するエミュレーターのソースプログラムには、その機能毎に当該機能に対応する名称(以下「機能名称」という)が記述されているものと仮定する。

【0074】まず、スイッチ手段は、上記第1の実施の形態1でも使用した属性管理テーブルT2を読み込んだ後、エミュレーターのソースプログラムを読み込む(図13、ステップ1→ステップ2)。もちろん、エミュレーターのソースプログラムを読み込んだ後、属性管理テーブルT2を読み込んで差し支えない。

【0075】ここで、当該エミュレーターは複数のハードウェアの動作をすべて網羅するように作成されているので、このエミュレーターが備えている機能のうち当該

ハードウェアが備えていない機能は無効にしなければならない。そこで、スイッチ手段は、上記のように読み込んだエミュレーターのソースプログラムから上記機能名称を検索(図13、ステップ3)し、この検索に該当した機能名称のうち属性管理テーブルT2に登録されていない機能名称を無効にする(図13、ステップ4)。

【0076】なお、機能名称を無効にするには、その名称部分をコメントアウトするなどコンパイル対象から外せばよい。もちろん、コメントアウトしなかった機能名称はコンパイル対象となるので有効である(上記第5の実施の形態においてアプリケーションモジュール2-2の機能を行効あるいは無効にする場合もこの手法を採用することができる)。

【0077】最後に、スイッチ手段は、このように書き換えられたエミュレーターのソースプログラムを出力し、当該書き換え処理を終了する(図13、ステップ5)。

【0078】以上のように、本実施の形態によれば、複数のハードウェアの動作をすべて網羅するように作成しておいたエミュレーターのソースコードを書き換えることによって特定のハードウェアの動作を再現したエミュレータープログラムを簡単に作成できるため、動作テストを行うときの生産性が向上する。

【0079】なお、上記スイッチ手段の具体的な態様については特に限定されるものではないが、アプリケーションの開発を支援するソフトウェアに当該スイッチ手段と同様の機能を備えておくのが好ましい。

(第7の実施の形態) アプリケーションの開発を支援するソフトウェア(以下「開発支援システム」という)において、個々の機能を実現するソフトウェアを部品のように扱い、これら部品を組み合わせて複数の機能を持つアプリケーションモジュール2-2を作成する手法が広く知られている。ここで、特定のハードウェア属性と一致する部品のみ使用可能とする制限ができれば、そのハードウェアにない機能をアプリケーションモジュール2-2に含めてしまうことがなくなるという利点がある。しかしながら、この制限機能を実現するためには、属性管理テーブルT2を参照しながら部品のリストを作成するという煩わしい作業が必要であった。

【0080】このような問題を解決するためには、上記図9と同様のロジックを持つフローチャートを使用し、すべての部品を含むリストと属性管理テーブルT2とを比較して当該属性管理テーブルT2に含まれる部品のみ使用可能とする機能(スイッチ手段)を開発支援システムに備えればよい。

【0081】また、種々のハードウェアでも万能に動作するようアプリケーションモジュール2-2を作成できれば、個々のハードウェア用に複数のアプリケーションモジュール2-2を作成する必要がなくなるためソフトウェア生産性が向上し、またソフトウェア管理も楽にな

る。しかしながら、通常はハードウェアによって備えている機能が異なるので、上記万能に作成されたアプリケーションモジュール2-2をそのまま動作させた場合は、当該アプリケーションモジュール2-2を動作させるハードウェアに無い機能まで操作パネルPに表示される不具合がある。

【0082】この不具合を解消するには、アプリケーションモジュール2-2が属性管理テーブルT2を参照するようにし、当該属性管理テーブルT2で有効な機能のみを選択的に実行するようなプログラムを備えればよい。このプログラムの機能選択方法は、上述した属性管理テーブルT2を使用した検索方法と同様である。

【0083】このようにすれば、アプリケーションモジュール2-2はその動作時に当該ハードウェアが備える機能のみを操作パネルP上に表示することになるため、ユーザーが混乱することなく利便性が高まる。

【0084】なお、あるソフトウェアの一部としてアプリケーションモジュール2-2が動作する構成を採用している装置では、母体となるソフトウェアにアプリケーションモジュール2-2が結合して動作する形態となるが、この形態も上述した形態と本質的には変わらない。すなわち、アプリケーションモジュール2-2は、母体となるソフトウェアの属性管理テーブルT2を参照して上記と同様の処理を実行するようになっている。

【図面の簡単な説明】

【図1】本発明を適用したデジタル複合機に組み込まれるソフトウェアの構成図。

* 【図2】本発明におけるスキャナ制御およびプリンタ制御のシーケンス図。

【図3】本発明を適用したデジタル複合機に組み込まれるソフトウェアの他の構成図。

【図4】ジョブ管理テーブルの内部構成例。

【図5】従来の操作パネル表示に関する説明図。

【図6】本発明における操作パネル表示に関する説明図。

【図7】操作パネルの表示状態を示す図。

【図8】属性管理テーブルの内部構成例。

【図9】本発明におけるアプリケーションモジュールの機能制限手順を示す図。

【図10】クラステーブルの内部構成例。

【図11】分類結果のフォーマット例。

【図12】操作パネルの表示状態を示す図。

【図13】本発明におけるエミュレーターの機能制限手順を示す図。

【図14】従来のスキャナ制御およびプリンタ制御のシーケンス図。

【符号の説明】

2-2 アプリケーションモジュール

2-3 スキャナコントロールモジュール

2-4 プリンタコントロールモジュール

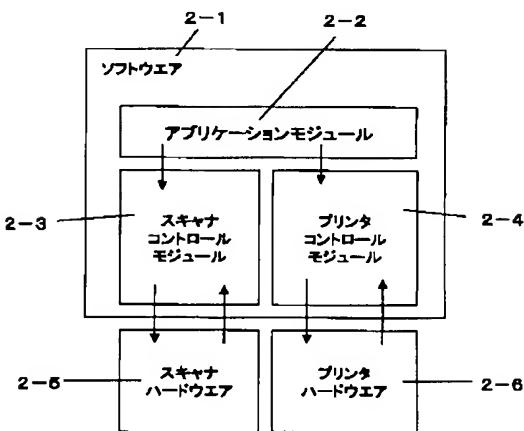
P 操作パネル

T 1 ジョブ管理テーブル

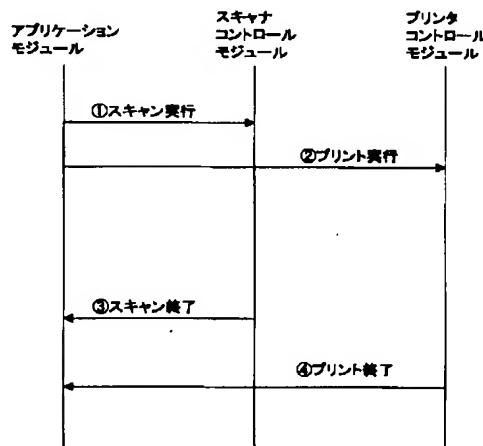
T 2 属性管理テーブル

T 3 クラステーブル

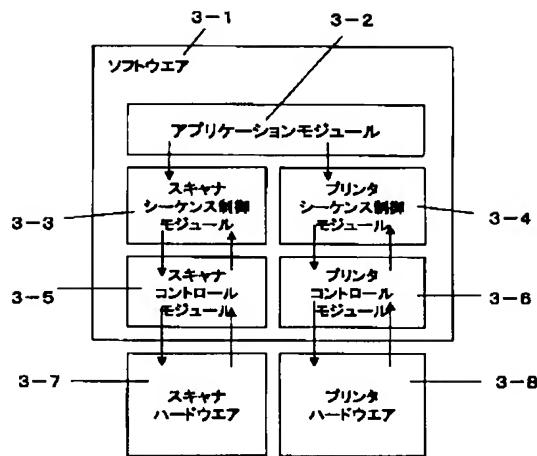
【図1】



【図2】



【図3】



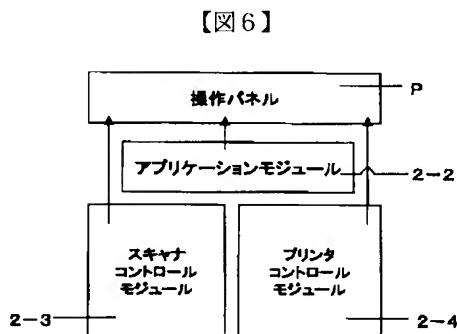
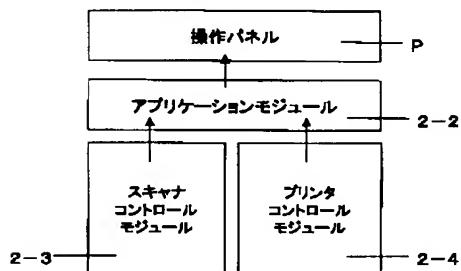
【図4】

名前	開始アドレス	状態	スキャン用ページ数	プリント用ページ数
T1	10x10000	スキャン実行	3	2
	20x10000	スキャン中	1	0

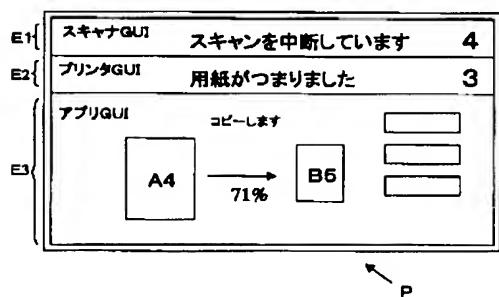
【図10】

名前	有無	値	値	値
両面プリント	有			
縮尺処理	無			
反転処理	無			
ページ選択	無			
濃度修正	有	5段階		
拡大縮小	有	1%単位	最小25%	最大400%
カラー	無			

【図5】



【図7】



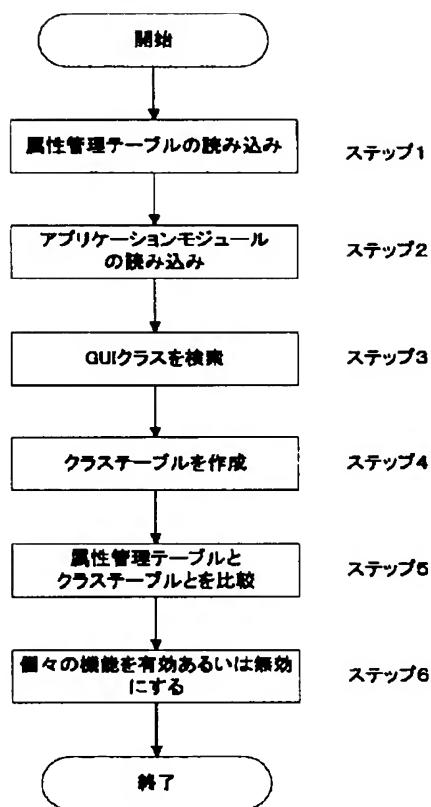
【図8】

名前	有無	値	値	値
両面プリント	有			
縮尺処理	有			
反転処理	無			
ページ選択	無			
濃度修正	有	5段階		
拡大縮小	有	1%単位	最小25%	最大400%
カラー	無			

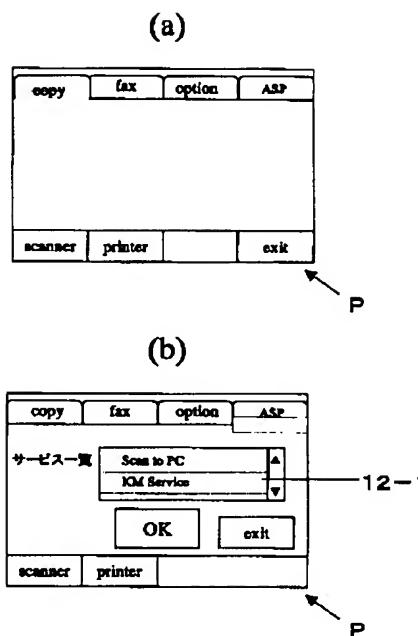
【図11】

共通	属性管理テーブルのみ	クラステーブルのみ
両面プリント	縮尺処理	反転処理
濃度修正		
拡大縮小		

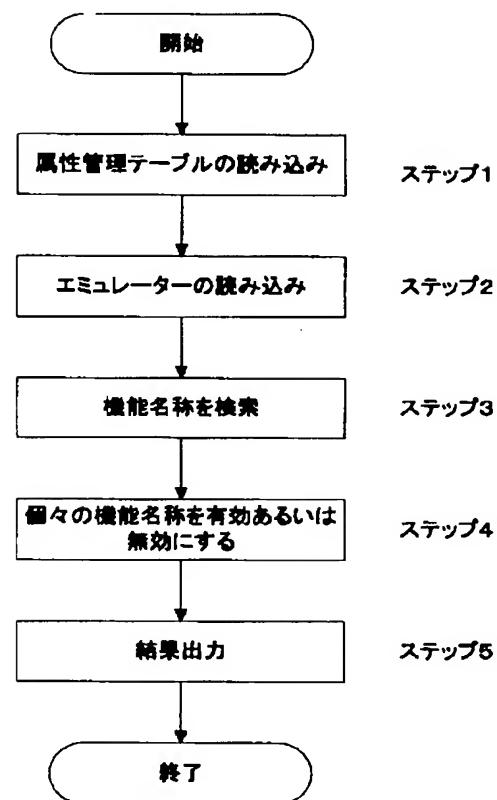
【図9】



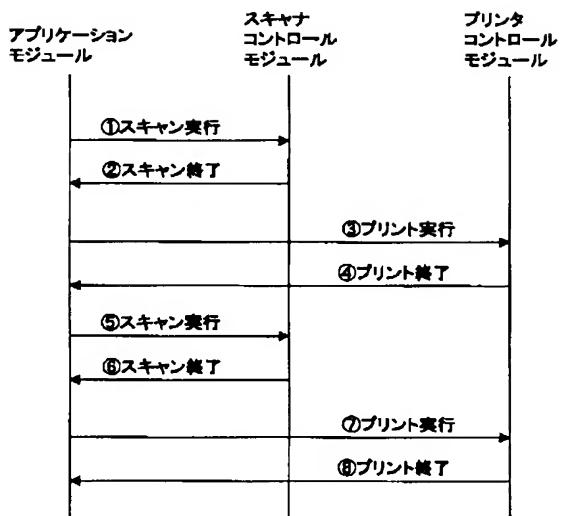
【図12】



【図13】



【図14】



フロントページの続き

(72)発明者 久富 健治 大阪府門真市大字門真1006番地 松下電器 産業株式会社内	F ターム(参考) 5B042 CA36 HH03 5B048 AA14 BB02 5B076 AA05 DD05 DD06
(72)発明者 松永 繁樹 大阪府門真市大字門真1006番地 松下電器 産業株式会社内	5E501 AA02 AA06 AA07 AC10 AC34 BA05 DA05